



Gameplay Analysis of Multiplayer Games with Verified Action-Costs

William Kavanagh¹ · Alice Miller¹

Received: 25 September 2020 / Accepted: 2 December 2020 / Published online: 23 December 2020
© The Author(s) 2020

Abstract

Measuring player skill cannot be done by considering their historical success alone as the relative skill of their opponents must be considered along with confounding factors such as luck and circumstance. With a specifically designed game, every possible player action can be attributed a cost, the value by which a player reduces their maximum probability of winning. By considering the costs of the actions made by a player we can obtain a more accurate representation of how skilful they are. We developed such a game, the mobile game RPGLite, and compared the actions players made with the cost values we had calculated. Through this analysis we made several observations about RPGLite which we share here to demonstrate the utility of action-costs for gameplay analysis. We show how they can be used to identify game states which players have difficulty making the best moves from, to measure how players learn over time and to compare the strengths and complexity of the characters of RPGLite. Commercial titles could benefit from similar tools—we discuss the feasibility of applying our approach to more complex games.

Keywords Gameplay analytics · Game design · Game balancing · Optimal play · Formal verification

1 Introduction

Traditionally player skill is measured by a ranking system such as the Elo rating system (Elo 1978) or Microsoft's TrueSkill (Minka et al. 2018). Some variations of these ranking systems encapsulate additional information from games beyond win/loss data, such as the margin of victory (Kovalchik 2020), the number of player kills (Ravari et al. 2017) or if the player is part of a group (DeLong and Srivastava

✉ William Kavanagh
w.kavanagh.1@research.gla.ac.uk

Alice Miller
Alice.Miller@glasgow.ac.uk

¹ Department of Computing Science, University of Glasgow, Glasgow, Scotland, UK

2012). Even then these are a compromised form of judging player skill, a better measure would be to consider every action a player makes and compare it to the best action they could have made in the same situation. This would not be possible for the majority of games where the variety of actions available and number of decisions a player makes are too vast. We have developed the game *RPGLite*¹ (University of Glasgow, 2020) as a game for which this is possible.

RPGLite was designed to be suitable for analysis whilst still offering players a challenging puzzle to solve. It is a two-player mobile game that operates through play-by-correspondence. Players take turns performing actions aiming to reduce their opponent's characters' health to 0. We calculate the *cost* of any action as the degree to which the acting player has reduced their optimal probability of winning. Consideration of the average cost of the actions performed by a player give an accurate accounting of the extent to which the player has "solved" the game, i.e. how good they are. We believe this to be a good indicator of player skill.

With a simple game like RPGLite it is natural to assume that players will converge upon an optimal strategy relatively quickly. One would expect players to make fewer mistakes in their 100th game than in their 1st, for example. However other experiments have generally failed to identify this behaviour (Levitt et al. 2010). Unlike in the games often studied, like card games or prisoner's dilemma type games, optimal strategies in RPGLite are not mixed, the optimal action(s) to take are the same every time a state is visited. This is different to Poker for example where players are rewarded for acting unpredictably (Ponsen 2008). We use action-costs to study if players learnt to play RPGLite over time by seeing if their average action-cost decreased as they made fewer mistakes and the mistakes they did make were less costly. We found that there were several players who actually got worse over time. We discuss possible causes for this.

Action-costs can be used in other forms of analysis beyond measuring player skill. They can also be used to inform judgements on any area where a game developer would benefit from knowing the frequency and extent of mistakes made. For example, we have used cost values to calculate the average error per action for characters in RPGLite, which we can then compare to identify which characters are more complex to play (so may require further explanation to players). This informs decisions on game balancing as a character may be effective enough, but they are being played poorly due to a lack of clarity in their design. Furthermore we identify where players made mistakes consistently and analyse the misconceptions made in so doing to gain a greater insight into how the game is perceived by players. We present our findings about RPGLite informed by action-costs as a demonstration of how other games could benefit from a similar form of analysis.

In this paper we first describe RPGLite in detail, the game, the configurations used and the mobile application developed for it. We then explain action-costs, the key contribution of this work, using example scenarios from RPGLite to describe how they are calculated. We then detail the gameplay analysis we performed on

¹ More information on the research RPGLite is part of and links to download the application itself can be found at <http://rpglite.app>.

RPGLite using action-costs, a study on player learning, a taxonomy of game characters and their complexity and an investigation of common mistakes. We then describe other uses for action-costs: to inform player rating systems and to educate players. Finally we discuss the impact of our work and its suitability to large, professionally-developed games.

In addition to the introduction of action-costs, our key contributions are: the development of a game setting which grants high confidence in the assessment of play and a study on player behaviour and aptitude in that setting.

2 RPGLite

2.1 The Game

The game that we use is an expansion of one of the same name that was created as a case study on game balancing (Kavanagh et al. 2019). RPGLite the game, is best described by its *rules*, *mechanics* and *configuration*.

Rules RPGLite is a two-player, turn-based game in which each player uses two characters from a pool of eight. Character selection occurs before a game begins without knowledge of opponent selection. Mirrored match-ups are possible where both players choose the same pair of characters. Each character has a unique action and three attributes: health, accuracy and damage. Some have additional attributes described by their action. On their turn, a player chooses the action of one of their alive characters (those with over 0 health) and targets an opposing alive character with their chosen action. That action will succeed or fail based on a dice roll of 0–99 and the acting character's accuracy value, if the roll is at least 100 minus the accuracy value then the action is successful. Players can choose to skip on their turn or to forfeit the game at any time. A coin is flipped to decide which player goes first and the winner is the player who is first to reduce both of their opposing characters health values to 0.

Mechanics The mechanics of RPGLite are encapsulated in the eight characters and their actions:

- *Knight*: targets a single opponent;
- *Archer*: targets up to two opponents;
- *Healer*: targets a single opponent and heals a damaged ally or themselves;
- *Rogue*: targets a single opponent and does additional damage to heavily damaged targets;
- *Wizard*: targets a single opponent and stuns them, preventing their action from being used on their subsequent turn;
- *Barbarian*: targets a single opponent and does additional damage when heavily damaged themselves;
- *Monk*: targets a single opponent and continues turn until a target is missed or a different action is chosen, and;
- *Gunner*: targets a single opponent, does some damage even on failed actions;

Table 1 Configurations used for RPGLite

Character	Attribute	Initial	Updated	Character	Attribute	Initial	Updated
Knight	health	10	10	Wizard	health	8	8
	accuracy	60	80		accuracy	85	85
	damage	4	3		damage	2	2
Healer	health	10	9	Gunner	health	8	8
	accuracy	85	90		accuracy	75	70
	damage	2	2		damage	4	4
	heal	1	1		graze	1	1
Archer	health	8	9	Monk	health	7	7
	accuracy	85	80		accuracy	80	75
	damage	2	2		damage	1	1
Rogue	health	8	8	Barbarian	health	10	9
	accuracy	75	70		accuracy	75	70
	damage	3	3		damage	3	3
	execute	5	5		rage damage	5	5
					rage threshold	4	4

The additional attributes needed to describe the characters fully are the *heal* value of the Healer, the heavily damaged value for the Rogue (the *execute range*), the heavily damaged value for the Barbarian (the *rage threshold*), the increased damage value for the Barbarian (the *rage damage*) and the miss damage (*graze*) for the Gunner.

Configuration In total there are 29 attributes for the characters in RPGLite. A configuration for RPGLite is a set of values for each attribute. We used two separate configurations, after collecting over 3000 games with the first, we stopped all games in progress and updated the game with the second. The *health* attribute is the amount of damage a character can sustain before no longer being usable, *accuracy* is the probability (as a percentage) that their actions will be successful and *damage* is the amount by which their actions reduce opponents health when successful. The configurations used are shown in Table 1. We tested the configurations to ensure they were sufficiently balanced before releasing them. When devising configurations we first ensured that no character had at least as much initial health as another whilst at the same time having at least the same expected damage as them.

These attributes are the parameters we tune in an attempt to balance the game. The application was released with a configuration which we suspected of being balanced based on automated analysis. After a significant number of games were played, the application was updated with a new configuration (dubbed “season two”), with the aim of maintaining player interest. The new configuration had altered attributes for seven of the characters, for example the Healer’s health value decreased from 10 to 9 and their accuracy increased from 0.85 to 0.9. Only the Wizard remained the same between configurations.

2.2 The Application

A mobile application was the obvious platform for RPGLite so a lightweight application of the same name was developed (Wallis et al. 2020). The design goals of the application were to encourage players to play competitively and to play a lot. The game is played by correspondence, when one player makes a move the other receives a notification and has 24 hours to make their move before their opponent can claim victory by default. In order ensure a high number of games were played, but that players remained invested, games are limited to 5 per user at a time and the application includes peripheral systems to incentivise play. These systems include a rating system, *skill-points*, which are weighted towards those who play more games and are shown on a leaderboard. Players can also earn medals, the requirements for which encourage playing many games or playing to win. e.g.: “Use the Knight 25 times” or “Win 5 games in a row”.

The application was made available on the Android and iOS app stores. It was promoted by the developers via social media as well as directly to fellow researchers. Four months after the initial release there were 354 registered users and 8671 completed games. Of the completed games, 6959 (80.25%) were completed successfully (where neither player abandoned or forfeited the game), 3546 of which (50.96%) occurred in “Season Two” after the configuration was updated. Only the successfully completed games from either season are considered for analysis. The accounts of the developers of the application and players who completed fewer than 5 games are also excluded from the analysis, leaving 134 users as the total participants (37.85% of registered users). Of the registered users 183 (51.69%) failed to complete a single game. In this paper we have omitted all usernames.

Game documents are stored in a database along with details of every move made, using the following notation: Players are allocated as either player 1 or player 2 and the character acting or being targetting is referred to by a capitalised initial. For example, the string “p1Kp2A_12” denotes to the action shown in Fig. 1 (right), or “player 1’s Knight targeted player 2’s Archer and rolled a 12”. Every action is accompanied by a single roll except when an Archer targets two opponents in which case there is a roll for each, e.g. “p2Ap1R_67p1A_56”, or when a player skips which requires no roll.

The dataset collected from the application is available freely online (Kavanagh et al. 2020), including documents for all completed games, all users and all key interactions with the application.

3 Action-Costs

3.1 Methodology

Individual games of RPGLite can be modelled as a two-player stochastic game (Shapley 1953) where players have a nondeterministic choice of action followed by a probabilistic transition determining the outcome. Actions either *hit* or *miss* in all cases except when the Archer targets multiple opposing characters, in which

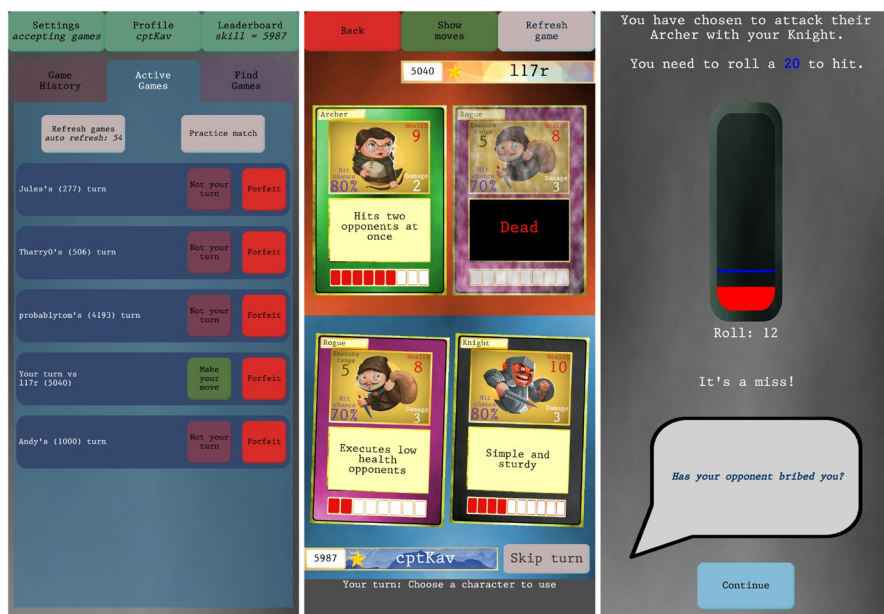


Fig. 1 Screen captures from RPGLite running on a mobile phone, from left to right: The home screen showing the five game slots, an active game where the player must choose which character to use, the action result screen after a roll animation has concluded showing an unsuccessful action.

case *hit_one* and *hit_other* are also possible, or when a player skips their turn. The Prism-Games model checker (Kwiatkowska et al. 2018) can be used to analyse these models in the form of a stochastic multiplayer game (SMG). For an SMG, Prism-Games can calculate the optimal value, the highest probability of reaching a subset of states that a player can obtain through their action decisions in a competitive setting. The optimal value is the maximum probability that the player can achieve their goal when they are trying to maximise their chances of achieving their goal, whilst their opponents are trying to minimise the player's chances of achieving this goal. This is also known as the minimax value. For RPGLite, the states in which a player has won are those where they have at least one character with over 0 health and their opponent does not. From this, we can obtain the optimal value for a player winning in any reachable state of the game.

States in RPGLite are encoded as a 19-tuple describing whose turn it is and the health values of either player's 8 characters as well as which are stunned:

$$s = (\text{turn}, p1\text{Knight}, p1\text{Archer}, \dots, p1\text{stun}, p2\text{Knight}, p2\text{Archer}, \dots, p2\text{stun})$$

Initial transitions involve a coin flip to set the turn and each player reducing the health of 6 of their 8 characters to 0, representing character selection. The stun values give the index of the character that is stunned. It is 0 when no character is stunned, 1 if the Knight is stunned, and so on. When calculating optimal values from

each state we only consider player 1's probability of reaching a winning state, this is sufficient because the game is symmetrical, any state player 2 can reach can be rewritten for player 1. When parsing actions for player 2 we rewrite the state and action as if for player 1, for the state this means setting turn to 1 and swapping the 9 variables for player 1's character state with those for player 2. We use Prism to generate state lookup tables giving the optimal probability of player 1 winning from any state. When generating the lookup tables we set the characters used by player 1 to each of the 28 pairs available and calculate the probabilities from each state reachable against any opposing material. Some examples from the Knight-Archer table are:

(1,8,4,0,0,0,0,0,0,0,0,0,2,6,0,0,0,0,0)=0.7454295392843602
 (1,8,4,0,0,0,0,0,0,0,0,0,2,8,0,0,0,0,0)=0.4651701837999699

This tells us that when it is player 1's turn and they have a Knight with 8 health, an Archer with 4, their opponent has a Wizard with 2, a Rogue with 6 and no character is stunned, player 1's optimal probability of winning is 0.745, whereas if the opposing Rogue had 8 health, it would be only 0.465.

Prism can also be used to perform strategy synthesis (Kwiatkowska 2016; Kavanagh et al. 2019; Giaquinta et al. 2018), detailing the optimal strategy for a player – the strategy which has the highest probability of winning against an opponent trying to minimise it. This gives a single action from every state which will maximise the player's optimal probability of winning, but it does not evaluate the extent to which any other action is incorrect. It also fails to account for states where multiple actions are optimal. This will happen in RPGLite, especially frequently when a player uses the Rogue-Barbarian pair as these two characters share similar actions and attributes.

By combining the optimal values at every state with the actions available and the probability of each action's success, we can generate a useful resource for play analysis, the action lookup table. Multiplying the probability of an action's outcome and the optimal value for the player at each outcome gives the player's optimal probability of winning having chosen an action at a state, before the outcome is known. The action lookup tables list states and the probabilities associated with each available action from the state, rather than the single optimal value shown in the state lookup tables. We have generated 28 such tables, one for each character pair. The values for the Knight-Archer states shown above are:

1,8,4,0,0,0,0,0,0,0,0,0,2,6,0,0,0,0,0:
 {skip:0.45563,K_W:0.49406,K_R:0.74543,A_W:0.49406,A_R:0.49671,A_WR:0.71668}
 1,8,4,0,0,0,0,0,0,0,0,0,2,8,0,0,0,0,0:
 {skip:0.22326,K_W:0.34959,K_R:0.43816,A_W:0.34959,A_R:0.40916,A_WR:0.46517}

The action with the highest value associated to it is the optimal action as it provides the highest optimal probability. Here we can see that when the opponent has 6 health on their Rogue, the best action is to attack it with the Knight (K_R) rather

than attack both opposing characters with the Archer (A_WR), whereas when the opponent's Rogue has 8 health the best action is to use the Archer. This example demonstrates the subtleties of the game – small changes in state lead to differing action viability. Note how the value associated with the optimal action is equal to the optimal probability at the state, as expected.

With the action lookup table we can calculate the *cost* of every action taken. This action-cost is calculated as the difference between the optimal probability for a player having chosen the move and the maximum optimal probability possible from all moves available at a state. In the example above, the cost of playing K_R when the opponent's Rogue has 8 health is $0.46517 - 0.43816 = 0.02701$, the player has reduced their optimal probability by almost 3% by not playing A_WR.

The state and action lookup tables for both configurations are included in the RPGLite dataset (Kavanagh et al. 2020).

3.2 Considerations

The *relative cost* of a move is calculated by dividing the cost by the optimal probability. The relative cost of the example above is $0.02701 \div 0.46517 = 0.05806$. Relative cost is important to consider because it contextualises the costs with the game state. A reduction in probability from 0.4 to 0.1 is more significant than from 0.8 to 0.5, cost does not reflect this, but relative cost does. We exclusively use relative cost in our analysis of RPGLite for this reason.

We found that a strong indicator of which player won a game was the number of high cost moves, rather than the cumulative cost. When analysing game data we consider these high-cost moves, and refer to them as *mistakes*. For example, we might classify a minor mistake to be a move with a cost of at least 0.1 and a major mistake to be a move with a cost of at least 0.25.

There are some game states where players are less likely to be motivated to win, the action-costs in these states should not be considered along with others. For example if a player is losing heavily and defeat is seemingly inevitable then they are likely to be less motivated to win. Similarly states where players have only a single action available to them (such as when players are forced to skip by a Wizard) should be discounted as the single action will be optimal so the cost will be 0.0. Nothing can be learned by considering these actions. We refer to moves made in states where players are motivated to win as *critical actions* which we define as states where more than one action is available and the player's optimal probability of winning is greater than 0.15.

4 Gameplay Analysis

4.1 Player Learning Over Time

We use action-costs in several ways to measure player learning—when examined over time they should show how player skill changes. We believe this gives a more

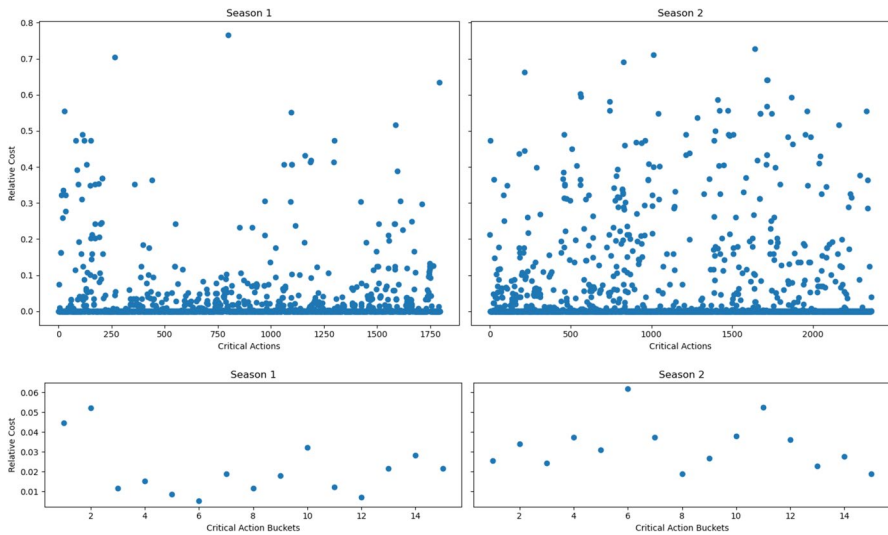


Fig. 2 (Above) the cost of all critical actions taken by a single user sorted chronologically by game end time. There are 1796 actions shown in season 1 with an average cost of 0.021 and 2363 in season 2 with an average cost of 0.033. (Below) the same data in 15 buckets for each season

accurate measure of skill than matchmaking ranks (Aung et al. 2018). As games are played, players will begin to see the affect of their actions (i.e. if they lead to success or failure) which should lead to their making better decisions. We hypothesised that as more games were played by players their average action-costs would decrease as they made fewer mistakes and those they did make would be less significant.

It is difficult to draw any conclusions from the action-costs of a player when considered as a whole. Action-costs for season 1 cannot be compared to those for season 2 as players reach different states in the two seasons, from which non-identical actions are available with different costs. Fig. 2 shows the relative costs of all actions taken by a single player from every critical state reached. Over 4000 actions are shown across both seasons. The data is sparse—the majority of actions taken were optimal, with a cost of 0.0. In total 73% of the player’s season 1 actions and 77.8% of their season 2 actions were optimal. There is no noticeable trend in the data of all actions or from taking averages throughout. There were more higher cost actions when the player first started playing which could show their initial learning, but this could also be down to the player’s material selection. The player used a Rogue-Monk pair for the majority of season 1 after some initial exploration and continued to use them for their first 50 games of season 2, but did not use them afterwards. The effect of this can be seen in the figure as the dense clustering of > 0.0 and < 0.1 cost actions from action 300 in season 1 up to action 300 in season 2.

Having studied all costs for a single player it is clear that action-costs must be used with more context. The actions available to a player are dependent on the game state, which itself is a factor of the material chosen by both players. We use mistakes, as described in Sect. 3.2, to counteract this. To verify the soundness of this approach we must ensure the possibility of making mistakes is similar with all materials. We

Table 2 A snippet of material comparison for all pairs involving a Knight in season 2.

Pair	critical actions	minor available	major available	avg. cost of 2nd best
KA	158648	0.849	0.291	0.08788
KW	144797	0.867	0.345	0.09070
KR	49695	0.866	0.331	0.10371
KH	215461	0.819	0.288	0.06013
KM	163431	0.904	0.244	0.04540
KB	72649	0.856	0.320	0.08824
KG	173258	0.875	0.318	0.08222
...

Critical actions is the number of critical states reachable against any opposing pair. For each pair shown the number of critical actions is given along with the proportion of those actions from which a minor mistake (> 0.1) and major mistake (> 0.25) can be made and the average relative cost of the second best action from all of those states

compared all possible actions for all possible pairs, some abridged results are given in Table 2 showing all pairs with a Knight. Across all pairs, the proportion of states from which a minor mistake is available ranges from 0.819 to 0.931 and for a major mistake it ranges from 0.216 to 0.413. The opportunities to make errors do not vary hugely between materials, which we believe allows for the use of mistakes at various thresholds as a unified measure across materials.

Fewer of our players exhibited a clear negative trend in average cost over time than we expected. To measure the rate of learning amongst the entire playerbase we considered the proportion of moves made in each game that were above a mistake threshold. Looking at a smaller number of games, the rate of learning is consistent, players gradually improve and make fewer mistakes per game. As the number of games considered approaches 100 the rate of learning slows to the point where mistakes are made at the same rate game to game. What is surprising is that after around 150 games had been played, the rate of mistakes being made increased. Fig. 3 shows this analysis performed considering major mistakes. The same trends of steady initial learning, flattening around 80–100 games and then players getting worse after 150 games, is apparent when considering minor mistakes and average costs. It is important to note that the populations being considered are not the same, as shown by the figure we had data on 87 players who had played at least 25 games, but only 19 who had played at least 200 so the data is more sensitive. The populations overlap, all players to have played 50 games are included in the data for those who played 25 for example, so results should be consistent throughout all groups.

From Fig. 3 it appears that the second game is played worse than the first for all four levels of experience and other early games appear to be worse than expected. This may be a symptom of the way characters in RPGLite are unlocked. New players can only use the Knight, the Archer, the Rogue and the Healer. After having finished at least one game with all of them they unlock the Wizard, completing a game with the Wizard unlocks the Barbarian. The Monk and Gunner are unlocked in the same way. It takes a minimum of five games to unlock all of the characters and it

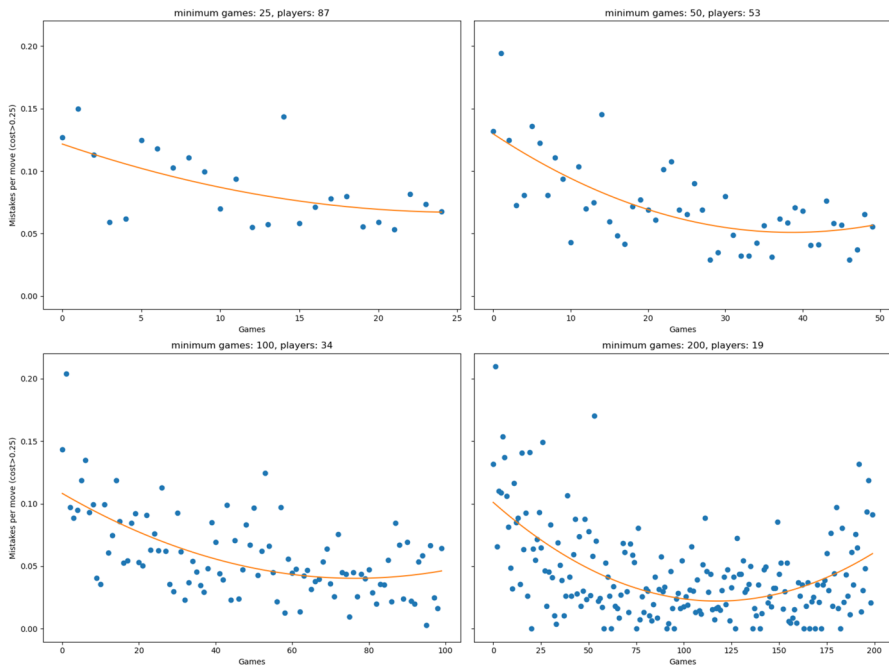


Fig. 3 Proportion of moves which were a major mistake per game. Shown for the first n games played by all players to have played at least n games, with values for n of 25, 50, 100 and 200. A quadratic fit is included to indicate trends

was a conscious decision that the characters unlocked at the start of the game were the most simple to understand, with more complex characters coming later. Players are also likely to explore the characters in early games and then settle on preferred materials in later games. Of the 53 players who played over 50 games only 21 used every character more than 5 times.

The increase in the rate of major errors being made by experienced players was not expected. We expected learning to flatten out when players had *solved* the game and then errors would only be the result of carelessness, but the results clearly show a decline. One possible reason for this is players growing apathetic with the simple nature of the game and exploring once again. This does rather contradict anecdotal evidence—many players claimed they were highly motivated by skill-points and the leaderboard—those who played the most were the ones competing for the top positions. Despite this decline in decision making, the average rate of major errors decreased in the final population of players when compared to those who played 100 games, as it did every time the number of games considered increased.

The most precise method for appraising player learning is to consider actions made in states that the player has visited before. Unlike our other approaches to studying player learning, this approach does not need to account for material or game state. However, a drawback is that many repeated states are likely to be those at the beginning of a game, from which the optimal action is easier to identify. A simple heuristic for choosing which action to take is to choose the action

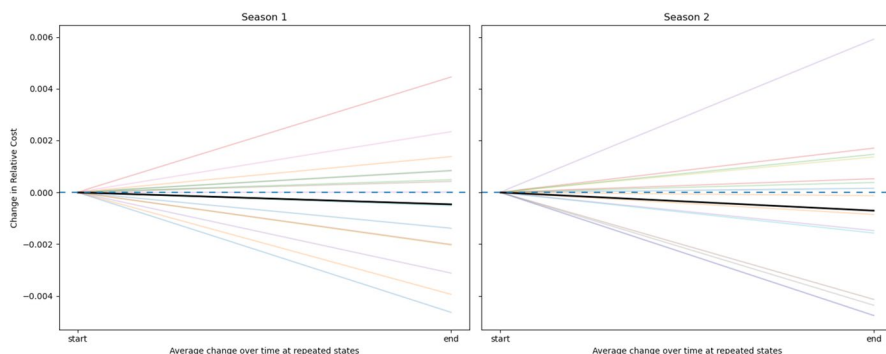


Fig. 4 The average change in cost for the top 15 players when considering only states visited multiple times. Bold lines represent the average in either season

from which more value is *expected* to be gained in terms of damage done to an opponent or undone to the player's own characters. The expected value is calculated by multiplying the beneficial change (opponents damaged and allies healed) in a reachable state by the probability of reaching that state and summing the values over those states. Consider the first move in a Knight-Archer v Knight-Archer game in season 2: The Knight does 3 damage to one target and can *hit* or *miss*, whilst the Archer does 2 damage to two characters and can *hit-both*, *hit-one*, *hit-other* or *miss*. Both characters have an accuracy of 80. The optimal action is to use the Archer as you can *expect* to deal damage with a value of $(0.8^2 \times 4) + (0.8 \times (1 - 0.8) \times 2) + ((1 - 0.8) \times 0.8 \times 2) + ((1 - 0.8) \times (1 - 0.8) \times 0) = 3.2$ whereas with the Knight you can only *expect* $(0.8 \times 3) + (1 - 0.8 \times 0) = 2.4$. Having chosen to use the Archer your optimal value is 0.6464, i.e.: if you and your opponent play perfectly from hereon your probability of winning is 0.6464, had you chosen the Knight it would be 0.57446 or 0.5488 if you had targeted the opposing Archer or Knight respectively. Now consider the same match up where all characters have only 3 health remaining, the value you can *expect* from your actions remains the same, but you could reduce an opposing character to 0 preventing their actions from being used – how much is this worth? Using action-costs the additional value of reducing an opponent character to 0 health, thereby preventing them from acting, can be calculated. In this state choosing “Archer attacks Knight, Archer” leads to an optimal value of 0.6494, “Knight attacks Archer” leads to 0.27649, but the optimal action of “Knight attacks Knight” leads to an optimal value of 0.74719. This decision is more complex due to the additional nuance of dealing with low-health characters. Knight Archer vs Knight Archer was played 644 times in season 2, all of which would have visited the first state, but there were no games which reached the state described above.

To determine if players improved we calculated the average change from every state that they visited multiple times. The results of this calculation being performed on the top 15 players in either season is shown in Figure 4. The average change in season 1 is -0.00046 and in season 2 is -0.00071 . The values are small because in the vast majority of states players played optimally every time, so there was no

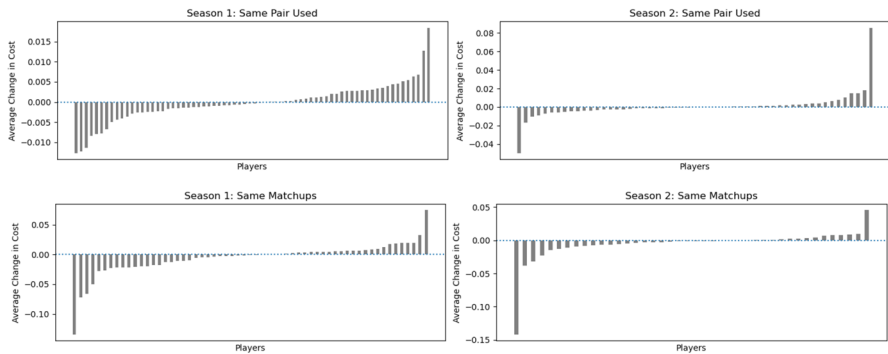


Fig. 5 Learning within pairs played and matchups experienced. Values below 0.0 denote a player that got better with experience, values above denote a player that got worse. Individual results have been sorted into ascending order

change. The results indicate that the top players got slightly better over time, on average, but there are several instances of players getting worse. Calculated for the top 75 players in either season, the averages are 0.00497 in season 1 (they got worse on average) and -0.00395 in season 2.

A notable finding from the costs in repeated states was the extent of player obstinacy. Many players made an error the first time they visited a state and then repeated the error on every future visit. Of the 20 players who played the most games, 18 repeated errors they made when visiting states for the first time in every subsequent visit more often than ever choosing another action. One player visited 159 states multiple times in which their initial move was a mistake, from 139 of those states they made the same incorrect move every time.

There are a limited number of states that players visit multiple times. Many of these are inconsequential in that they are solved at the player's first visit. The states which are typically more complex are rarely seen. Too few games were played for us to achieve very accurate results by just considering states visited multiple times. We broadened the criteria from states that a player has seen multiple times to material they choose frequently. We expanded on this too, to consider matchups (the combination of characters used by both the player and their opponent) that they play repeatedly.

Many players only used a limited subset of the available material, sometimes choosing to play with the same pair of characters to every time. We considered the change in their average cost over time within those material choices, suspecting that players who restrict themselves to only some of the pairs will get better at using those pairs, selecting lower cost actions. To study pair-wise learning we consider each season separately, disregarding players who played fewer than 10 games, and find the average trend in mean cost per game using each pair played at least 5 times. Our results are shown in Figure 5 (top). In both seasons the numbers of players who improved and players who got worse are similar, 38 out of 70 in season 1 and 30 out of 55 in season 2. The average across all players when weighted by games played in both seasons is between 0 and -0.00001 which we consider to be insignificant.

Figure 5 (bottom) shows results for analysis considering matchups players experienced at least 3 times in a season. The results are similar to the pair-wise analysis, 33 out of 59 players improved in season 1 and 27 out of 42 improved in season 2. The average learning restricted to matchups when weighted by games played is -0.00124 in season 1 and -0.00104 in season 2. Players did get slightly better with experience on average. However, several players got worse in both seasons when considering matchups.

Another indicator we looked at for the rate of learning was the point in a sequence of similar games at which players played worst. Following on from the pair-wise and matchup-wise analysis, we looked at every player with over 20 games in a season and calculated the average position in a sequence of at least 10 games played by the same pair or at least 5 featuring the same matchup in which the player made the highest average cost per move. We expected that the players' worst game would come early in these sequences. Of 137 sequences of games played by a player using the same pair in season 1, the average sequence was 31 games long, the worst game coming on average at the 14th position. Of 115 sequences in season 2 the average length was 9 and the worst game was the 4th. Similarly for sequences of matchups, of 196 in season 1, the average length was 39 and worst game was the 19th and of 241 in season 2, the average length was 12 and the worst was the 5th. All of these worst games came at a point between 40 and 50% of the length of the sequence. If significant learning was taking place one would expect the worst game to come much earlier in the sequence of similar games.

Having used costs in the analysis of player learning we can show that over half of all RPGLite players did improve over time, but several did not. We hypothesised that owing to RPGLite's simplicity some players would essentially *solve* the game leading to them incurring minimal costs on actions across several games, which was not the case. Costs are a good measure for analysing player skill. However in order to track learning the various actions and states available have to be accounted for. Costs can also show surprising patterns in player learning, as illustrated by the eventual drop-off in success we identified amongst keen players.

4.2 Material Comparison

When designing RPGLite's characters we wanted to ensure each was unique. The Knight is the basic framework that the others are built upon, intended to be simple. The Gunner and Healer are similar in that they are consistent throughout a game. The Wizard, Rogue and Barbarian all get stronger as the game proceeds: the Wizard can repeatedly stun a single target stopping a player from acting if they already have a character at 0 health and the others get more effective at lower health values. The Monk and Archer are often better early on when a character cannot be reduced to 0 in a single hit. Some characters combine especially well. The Rogue-Monk pair is particularly effective because the Monk can reduce characters to 5 hit-points, setting the Rogue up to *execute* them, all in a single turn. This combination was identified by many players in Season 1, being played 1046 times compared to the second most popular pair Wizard-Gunner which was played 624 times.

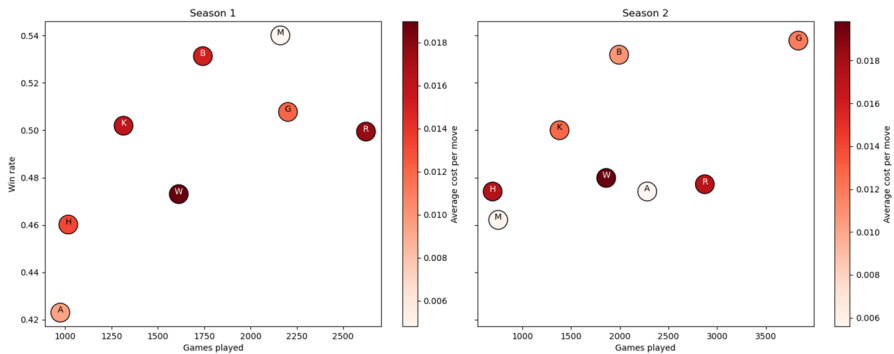


Fig. 6 Character popularity, usage and complexity in both seasons

Gameplay analysis to consider game balance often consists of pick-rate, win-rate calculations which consider lone material units. Using action-costs we can improve on this to get more information on how the characters are played and suggest which are confusing to players. In Fig. 6 we present a pick-rate, win-rate graph that illustrates how well the characters were played. In season 1 for example the Wizard was the worst played character, whilst the Monk was the best played. This additional information can be used by developers to make better judgements on the state of game balance. The Archer in season 1 was the least successful and least popular character. One possible reason for this could be that players were not good at using the Archer effectively. However knowing that the average cost of Archer actions was among the lowest of all characters suggests this is not the case. For season 2 we improved the Archer's attributes (increasing their health from 8 to 9 and slightly reducing their accuracy from 85 to 80) on the basis that it was not the fault of the players that the Archer was under-performing. We made a similar decision in reverse for the Monk, observing that players were making very few errors whilst winning consistently. These characters saw the most significant shift in popularity and success between seasons, even though five others were also modified. There was a general decrease in costs between seasons, which is to be expected as the majority of second season players were experienced – having played in the first season. We assumed that the Wizard would see a reduction in average cost, but it remained constant between seasons as did the Wizard pick and win rates (Fig. 7).

Developers of games where players or teams use sets of material rather than individual units often cannot easily weigh up the pros and cons of each of those sets because there are too many of them. RPGLite was designed with this analysis in mind, having a set size of only 2—the pairs that players select. This is small enough to allow for pairwise analysis, in addition to the material units themselves. The analysis of the full sets players use is a better source of data on the state of the game than analysis of individual units, but those sets cannot be changed directly to try and improve the game, instead the individual units can be adjusted, which affects all sets in which they are included. This is a significant factor in the difficulty of balancing games where material sets are used.

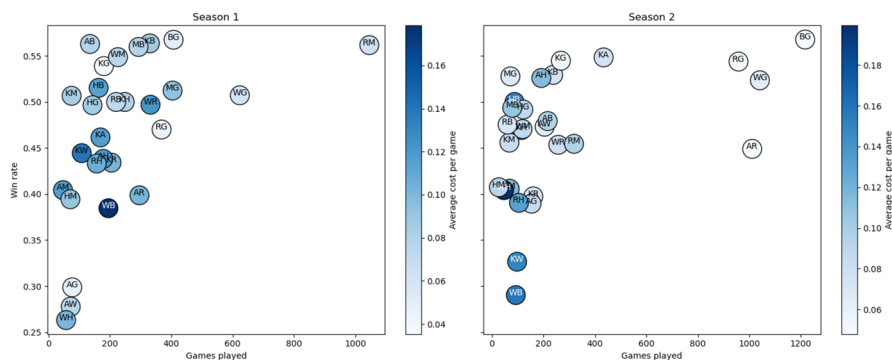


Fig. 7 Pair popularity, usage and complexity in both seasons. Names are abbreviated so KA refers to a Knight-Archer pair. Obscured in season 2 is the pair with the highest average cost, Wizard-Healer

The measures which can be taken from RPGLite data relating to the pairs being played is different to that relating to characters alone. Whereas with characters we can calculate the average cost of every move made using those characters, we can't do the same for pairs as actions have a single acting character. Instead the average total cost of moves made in a single game by a player with each pair is used. E.g.: Players using the Wizard-Barbarian pair in season 1 made moves that reduced their optimal probability of winning by 18% in total, on average.

Exhaustive evaluation of the state of RPGLite's balance is beyond the scope of this paper. However, there are some notable observations from the cost data in relation to the popularity and success of RPGLite material. There is no clear correlation between the costs of using material and their success, although it appears that the most played pairs have a lower average cost. In season 1 the Archer-Gunner pair was played well as shown by the very low average cost per game, but it was among the least successful with a very low win rate of 30%. This is a clear indicator that the AG pair in season 1 is not an effective one, but is not enough evidence on its own to suggest changing either character. After all, the Archer-Barbarian and Barbarian-Gunner pairs are among the most successful with success rates over 55%. These discrepancies in the viability of sets made up of similar material units is a desirable feature of games as it suggests that some materials complement others, enhancing their viability. This implies that a deeper system of relationships between the materials exists and is one that players will need to comprehend to be successful. Without costs we could not be as confident in our assessment of the strengths of the various material in RPGLite.

4.3 Common Mistakes

Being able to find the states where players are making mistakes is beneficial to understanding how players interpret the game and where design is not as intuitive as it could be. It is also interesting to see the positions from which players frequently fail to play optimally.

Table 3 Moves from selected states in season 1 in an RM-BM matchup.

p1R	p1M	p2B	p2M	p1Rp2M	p1Rp2B	p1Mp2M	p1Mp2B	p1_skip
8	3	5	10	0.36852	0.39352	0.52907	0.59575	0.26209
8	3	5	9	0.39041	0.4198	0.59999	0.67917	0.28295
8	3	5	8	0.77822	0.6146	0.72196	0.68947	0.49653
8	3	5	7	0.68002	0.53595	0.67947	0.73771	0.43733
8	3	5	6	0.69415	0.56233	0.74548	0.8128	0.46542
8	3	5	5	0.89965	0.76788	0.85306	0.79683	0.65396

Bold values represents the optimal action to take at each state

The ‘skip’ feature of RPGLite was implemented to allow the model to navigate from states where one player had a single stunned character alive. In pre-release testing feedback we were asked why skipping wasn’t automated as there was no situation where a player would want to skip if they could take a character action. This is false however, there are states from which skipping is preferable to character actions. In season 1, the two states visited at least 5 times with the highest average cost per move were when only the Barbarian was alive for either player, one with 10 health and the other with 7. Because the Barbarian does 5 damage when at 4 health or less and 3 otherwise, if you cannot win in 2 successful actions when above 4 health then it is preferable to wait until your opponent has hit you so that you can. When at 7 health with an opponent at 10 your optimal probability of winning if you skip is 0.64624 and is 0.39296 if you attack. In the reversed state the values are 0.60704 if you skip and 0.37376 if you attack. No players skipped in these states even though they were visited 64 times.

A more detailed example from season 1 that shows the complexity of calculating optimal actions when considering the Barbarian’s *rage damage* is shown in 6 consecutive states detailed in Table 3. Here a Rogue–Monk player is acting against a Barbarian–Monk player and as the health of the opponent’s Monk decreases, the optimal action changes both in terms of the acting character and the target. The state where the opponent’s Monk has 10 health was visited 11 times in season 1 and the average cost of all of those moves was 0.14213, the third highest of any state visited more than 5 times.

Action costs trivialises identifying poor decision making from the players. Here we have also used them to highlight interesting optimal actions by looking at states at which players slipped up often. This gives a better understanding of the mathematics underlying RPGLite.

5 Uses Beyond Analysis

5.1 Cost as a Ranking System

RPGLite has its own ranking system in the form of skill-points. Skill-points were designed to favour players who played more games—far more points are gained by

winning (35–45) than are lost by losing (5–15) and it is not possible to fall below multiples of 100 unless repeated games are forfeited. This system was not designed to be an accurate measure of player skill, rather an incentive for players to play more games to climb up the leaderboard. When developing RPGLite we included logic to keep a record of the Elo ranking of all players, but we did not make this value visible to players and intended to use it solely for analysis. To our surprise Elo proved to be a poor indicator of success. This is likely because the skill ceiling is far lower in RPGLite than it is in Chess, the game for which Elo was devised. In fact Elo proved to be less effective than even the skill-points ranking when compared to player win ratios.

To create a better system for ranking players we implemented a simple procedure in which we calculated the average cost that each player's actions deviated from the mean cost of all players at that state. This compensates for the various costs available at different states and the use of different materials. We call this *cost deviance per move* and would expect a player with a lower cost deviance per move to be a higher-skilled player than one with a higher cost deviance. Clearly the average cost deviance per move is 0.0, denoting a player who made an average cost action at every state. For simplicity, any player with a negative cost deviance can be considered *good*—they have understood the game better than most—and any player with a positive cost deviance can be considered *bad*.

When comparing the three ranking systems and their relation to a player's win ratio, as in Figure 8, the effectiveness of cost as a ranking system is clear to see. Elo seems to have minimal relevance as a predictor of player success, whereas skill-points appears to be a good predictor, although that may be confounded by experienced players typically having higher skill-points.

5.2 Cost as a Teaching Tool

Games increasingly allow players to analyse their own performance after having played. This takes several forms, detailed statistics shown on profile pages, video replays of games played and League of Legends gives letter-grades for player performance (Novak et al. 2019). These systems incentivise and support high-level competitive play, a motivation that is regularly desired by game developers. Automated analysis that can grade player performance is expensive and often the calculations are kept secret to prevent players from gaming the system. With the costs from RPGLite it would have been very easy to implement a similar system.

As RPGLite was designed in part to study player behaviour and learning, we did not want to condition our players or steer them in any particular direction. Were this not the case RPGLite games could have ended with an analysis screen showing players all moves made by them and their opponent, detailing the cost of each move and, where the optimal move wasn't played, what action they should have taken. As professional gaming continues to rise in prominence, the demand for automated analysis will increase (Taylor 2020). Techniques similar to calculating and sharing action-costs are a good way of providing these tools to players, showing where mistakes were made and what improvements could have been made.

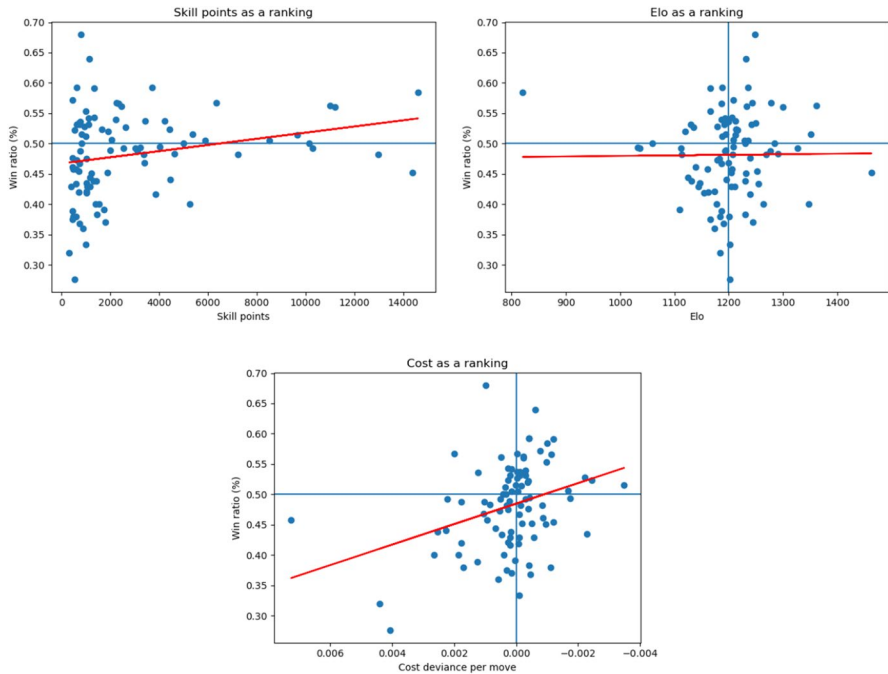


Fig. 8 Comparison of three ranking systems against the win ratios of all players with over 20 games played. Linear fit indicates correlation

6 Discussion and Conclusion

Action-costs can be used to track moment-to-moment decision making in intricate detail for all RPLite players and opens up many avenues for studying how the game is played. They show us that despite the game's lack of complexity there were no players who played perfectly. Our research on action-costs in the areas of game development and gameplay analysis is early work. These ideas could be expanded upon to learn more about how games are played and how better games can be developed.

6.1 Limitations

The findings from this paper are based on logged data from a single game. The identity of the players was kept anonymous which precluded any player profiling to separate those with prior experience of similar games. This experience would have likely affected initial skill and the rate of learning. Owing to how the game was advertised it is likely that the majority of the players were science and engineering postgraduate research students or had signed up to games research newsletters. This may have affected the way the game was played.

It is difficult to calculate the optimal values which inform action-costs and a series of compromises were made in the design of RPGLite to ease this. A comment we received in the early testing of the mobile application was *“having a 2 card deck makes it too heavily weighted towards luck over strategical use of your troop’s attacks/abilities.”* This comment is valid in that roll results and winning the coin-flip to go first are some of the most important factors for success in RPGLite, arguably more so than it would be in an equivalent, professionally-developed game.

RPGLite is an expansion on a case-study with only 3 characters. Ideally the teams would be triples, not pairs, but this lead to too great an increase in the processing required. For a candidate configuration of RPGLite such as the two used in the mobile app, we can calculate action-costs from all reachable states in around fifteen minutes using a standard desktop PC with 16GB of RAM. The character actions were also limited, common effects from other games such as damage on subsequent rounds or temporarily affecting the attributes of targets (e.g. lowering accuracy for n round) were discounted to limit state explosion (an increase in the size of tuples describing states and a subsequent exponential blow-up in the number of possible states). The health values of characters were kept low, although increasing them would have had little impact on the analysis. Decisions tend to be more impactful later in games so an increase in health values would have resulted in longer games, possibly reducing the size of the dataset we were able to collect. A more simple game makes the analysis of player behaviour more straightforward as we can better understand the mistakes made and misconceptions of the players. This likely differs from the design philosophy of games not created for research purposes.

Action-costs in asymmetric games such as RPGLite must be contextualised. Game material which significantly differs in what it allows the player to do is a design aim of many games, but this leads to differing actions and thereby differing action-costs being available to the players. Comparison between actions made using different materials is not always indicative of player skill. We have looked at ways of minimising the effect of this through limiting the compared actions to identical states or to similar material and through the use of mistake thresholds.

The use of action-costs as a measure for player skill is predicated on players being motivated to win. We are not professional game developers nor psychologists and only implemented measures which we believed would promote competitive play as well as explicitly describing the purpose of the application and how playing to win was required in greeting messages on the app itself. Some of our results suggested that players began to deviate from this behaviour after a large number of games as their action-costs increased. There is no reason to suggest that this lackadaisical quality to play would not be common in larger games.

6.2 Feasibility at Scale

It would be possible for action-costs to be calculated for larger and more complex games, however a more realistic objective for the immediate future may be for alternative values to be used rather than the minimax values, which require automated analysis of the entire state space. For example, historical data of a player or team’s

success from a given position could be used to estimate their probability of winning from that position. These values could be used in a similar way to how we use optimal values in our action-cost calculations. Another method for tackling the state space of more complex games would be to use abstraction (Johanson et al. 2013; Sandholm 2015) where some precision would be traded for more simple state descriptions, or the exploitation of symmetry to avoid searching redundant parts of the state-space, a technique commonly used in model checking (Miller and Calder 2006; Donaldson et al. 2009). Alternatively only specific scenarios could be considered to gain insight into problem areas of a game's design, limiting the amount of computation required.

Action-cost analysis is particularly suited to simple turn-based games, which have long been popular. There is already a significant amount of research into turn-based board games and card games which can support further work in the area. The relative success of RPGLite in terms of the number of players and games recorded shows the viability of games which can be analysed in this way as games which players *want* to play. As automated processing capabilities increase, the limit on the complexity of games which are enjoyable for humans to play remains constant.

Acknowledgements Our work creating RPGLite would not have been possible without the input of several colleagues and friends in the design and as participants. However, without Tom Wallis the application would not exist. He developed the application with us, implementing the network infrastructure and several key features which contributed to the success of this work. This work is supported by the EPSRC Doctoral Training Partnership award EP/M508056/1.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aung, M., Bonometti, V., Drachen, A., Cowling, P., Kokkinakis, A. V., Yoder, C. & Wade, A. (2018). Predicting skill learning in a large, longitudinal moba dataset, In: *2018 IEEE conference on computational intelligence and games (CIG)*, pp. 1–7.
- DeLong, C., & Srivastava, J. (2012). Teamskill evolved: Mixed classification schemes for team-based multi-player games, In: *Pacific-Asia conference on knowledge discovery and data mining*, Springer, pp. 26–37.
- Donaldson, A., Miller, A., & Parker, D. (2009). Language-level symmetry reduction for probabilistic model checking, In: *Proceedings of the 6th international conference on the quantitative evaluation of systems (QEST'09)*, pp. 289–298.

- Elo, A. E. (1978). *The rating of chessplayers, past and present*. Milton: Arco Publications.
- Giaquinta, R., Hoffmann, R., Ireland, M., Miller, A., & Norman, G. (2018). Strategy synthesis for autonomous agents using PRISM, In: *Proceedings of the 10th NASA formal methods symposium (NFM)*, pp. 230–236.
- Johanson, M., Burch, N., Valenzano, R., & Bowling, M. (2013). Evaluating state-space abstractions in extensive-form games, In: *Proceedings of the international conference on Autonomous agents and multiagent systems*. International foundation for autonomous agents and multiagent systems, 271–278.
- Kavanagh, W. J., Miller, A., Norman, G., & Andrei, O. (2019). Balancing turn-based games with chained strategy generation. *IEEE Transactions on Games*. <https://doi.org/10.1109/TG.2019.2943227>.
- Kavanagh, W. J., Wallis, W. & Miller, A. (2020). Rpglite player data and lookup tables, <http://researchdata.gla.ac.uk/1070/>.
- Kovalchik, S. (2020). Extension of the Elo rating system to margin of victory. *International Journal of Forecasting*, 36(4), 1329–1341. <https://doi.org/10.1016/j.ijforecast.2020.01.006>.
- Kwiatkowska, M., Parker, D., & Wilsche, C. (2018). PRISM-games: verification and strategy synthesis for stochastic multi-player games with multiple objectives. *STTT*, 20(2), 195–210.
- Kwiatkowska, M. Z. (2016). Model checking and strategy synthesis for stochastic games: From theory to practice. In: *Proc. 43rd International colloquium on automata, languages, and programming (ICALP 2016)*, pp. 4:1–4:18. <https://doi.org/10.4230/LIPIcs.ICALP.2016.4>.
- Levitt, S. D., List, J. A., & Reiley, D. H. (2010). What happens in the field stays in the field: Exploring whether professionals play minimax in laboratory experiments. *Econometrica*, 78(4), 1413–1434.
- Miller, A., & A.F., Donaldson. & Calder, M., (2006). Symmetry in temporal logic model checking. *ACM Computing Surveys*, 38(3), 8-es.
- Minka, T., Cleven, R., & Zaykov, Y. (2018). TrueSkill 2: An improved Bayesian skill rating system, Technical Report MSR-TR-2018-8, Microsoft.
- Novak, A. R., Bennett, K., Pluss, M., & Fransen, J. (2019). Performance analysis in esports: Part 1-the validity and reliability of match statistics and notational analysis in league of legends. Preprint at <https://osf.io/preprints/sportrxiv/sm3nj/>.
- Ponsen, M. (2008). The dynamics of human behaviour in Poker, In: *Proceedings of the Belgian-Dutch conference in artificial intelligence*, pp. 225–232.
- Ravari, Y. N., Spronck, P., Sifa, R., & Drachen, A. (2017). Predicting victory in a hybrid online competitive game: The case of destiny., In: *AIIDE*, pp. 207–213.
- Sandholm, T. (2015). Abstraction for solving large incomplete-information games, In: *AAAI Conference on Artificial Intelligence*.
- Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 39(10), 1095–1100.
- Taylor, N. (2020). The numbers game: Collegiate esports and the instrumentation of movement performance, In: *Sports, Society, and Technology*, Springer, pp. 121–144.
- Wallis, W., Kavanagh, W. J., Miller, A., & Storer, T. (2020). Designing a mobile games to generate player data—lessons learned, In: *GAME-ON 2020*, pp. 13–15. Available at <https://eprints.gla.ac.uk/223600/>.